REVIEW SENTIMENT ANALYSIS USING ML ALGORITHM

PROJECT REPORT

submitted in partial fulfilment for the requirement of the Degree of

BACHELOR OF SCIENCE (COMPUTER SCIENCE)

Submitted by
BARANI S
(20BCS007)

SRI KRISHNA
INSTITUTIONS
COIMBATORE

Under the Guidance
Prof. A.JEEVARATHINAM M.Sc.,M.Phil.,B.Ed.,(Ph.D.),
Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE

SRI KRISHNA ARTS AND SCIENCE COLLEGE
COIMBATORE 641 008

MAY 2023

REVIEW SENTIMENT ANALYSIS USING ML ALGORITHM

PROJECT REPORT

submitted in partial fulfilment for the requirement of the Degree of

BACHELOR OF SCIENCE (COMPUTER SCIENCE)

Submitted by

BARANI S

(20BCS007)

Under the Guide

Prof.A.JEEVARATHINAM M.Sc.,M.Phil.,B.Ed.,(Ph.D).,

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE

SRI KRISHNA ARTS AND SCIENCE COLLEGE

COIMBATORE - 641008

MAY 2023

Sri Krishna Arts and Science College
Kuniamuthur, Coimbatore -641008

DECLARATION

I hereby declare that the Project report entitled ***"Review Sentiment Analysis using ML Algorithm"*** submitted in partial fulfilment of the requirements for the award of degree of **Bachelor of Computer Science** is an original work and it has not been previously formed the basis for the award of any Degree, Diploma, Associate ship, Fellowship or similar titles to any other university or body during the period of my study.

Place: Coimbatore

Date: 06.05.2023

S. B

Signature of the Candidate

SRI KRISHNA
INSTITUTIONS
COIMBATORE

Sri Krishna Arts and Science College
Kuniamuthur, Coimbatore -641008

CERTIFICATE

This is to certify that the Project Report entitled "**Review Sentiment Analysis Using ML Algorithm**" in partial fulfilment of requirements for the degree of Bachelor of Computer Science is a record of bonafide work carried out by **BARANI S(20BCS007)** and that no part of this has been submitted for the award of any other degree or diploma and the work has not been published in popular journal or magazine.

06/05/23
GUIDE

6/5/23
HOD

DEAN

Dr. C. SUNITHA
Professor and Head
Department of Computer Science
Sri Krishna Arts and Science College
Kuniyamuthur, Coimbatore - 641 008

Dr. C. SUNITHA
Professor and Head
Department of Computer Science
Sri Krishna Arts and Science College
Coimbatore - 641 008

This Project Report is submitted for the viva voce conducted on 16.05.2023 at Sri Krishna Arts and Science College.

16/05/23
Internal Examiner

R. S 16/5/23
External Examiner

ACKNOWLEDGEMENT

The satisfaction and euphoria of the successful completion of any task would be incomplete without the mention of the people who made it possible, whose constant guidance, encouragement, crowned my effort with success.

I have been fortunate enough to secure co-operation, guidance and assistance from a number of people. I am at a loss of how to express the deep sense of gratitude I have towards all of them.

I convey my profound gratitude to **Dr.Sundararaman, M.com., M.Phil., Ph.D., CEO,** Sri Krishna Institutions for giving me this opportunity to undergo this project.

I am greatly indebted to our principal **Dr.R.Jagajeevan, MBA, M.Phil., Ph.D.,** and our Dean **Dr. K. S. Jeen Marseline, MCA.,M.Phil., Ph.D.,** who has given permission for the fulfilment of the venture.

I would like to express my gratitude to **Dr.C.Sunitha, MCA, M.Phil.,Ph.D.,** Associate Professor & Head, Department of Computer Science, Sri Krishna Arts and Science College for her invaluable support and guidance throughout my training.

I would like to extend my thanks and unbound sense for the timely help and assistance given by **Prof.A.JEEVARATHINAM, M.Sc.,M.Phil.,B.Ed.,(Ph.D).,** Assistant Professor, Department of Computer Science, Sri Krishna Arts and Science College in completing the training. Her remarkable guidance at every stage of my project was coupled with suggestions and motivation.

I owe my gratitude and thanks to **Mr. R. Thirumal, Senior Manager**, **NK International India Pvt Ltd** for permitting me to carry out the training.

I would like to express my sincere thanks to the God Almighty for his constant love and grace that he has bestowed upon me.

Finally, I thank my parents, family members, and my beloved friends for their moral support and encouragement without which I would not have been able to follow my dreams.

ABSTRACT

Sentiment analysis or opinion mining is the computational study of people's opinions, sentiments, attitudes, and emotions expressed in written language. It is one of the most active research areas in natural language processing and text mining in recent years. Its popularity is mainly due to two reasons. First, it has a wide range of applications because opinions are central to almost all human activities and are key influencers of our behaviours. Whenever we need to make a decision, we want to hear others' opinions. Second, it presents many challenging research problems, which had never been attempted before the year 2000. Part of the reason for the lack of study before was that there was little opinionated text in digital forms. It is thus no surprise that the inception and the rapid growth of the field coincide with those of the social media on the Web. In fact, the research has also spread outside of computer science to management sciences and social sciences due to its importance to business and society as a whole. In this talk, I will start with the discussion of the mainstream sentiment analysis research and then move on to describe some recent work on modelling comments, discussions, and debates, which represents another kind of analysis of sentiments and opinions.

CONTENT

CHAPTER 1

1. INTRODUCTION

1.1 ORGANIZATION PROFILE

Name of company :	NK International
Type of company :	Textile industry
Year formed :	1997
Office address :	n k international aa 1/3,sipcot industrial complex, pudukkottai-622 002, Tamil Nadu, India.
Contact person :	Mr. R. Thirumal – Managing Director.
Business nature :	Own production
Company website :	https://www.nkihometex.com/
Markets :	domestic and export markets.
Certification :	ISO 14001 certification

Genesis and evolution:

Started in 1997 to meet the demand for home textiles in the gulf countries, NKI has grown into one of the premier exporters of home textiles, eager to expand other markets all over the globe. The company obtained ISO 9001:2015 Certification in Quality Management Systems & ISO 14001:2015 Certification in Environment Management Systems. The officers of the Labour Welfare Department those of the EPF and other statutory inspection teams have appreciated our eco-friendly, worker-friendly measures. We have scrupulously acted on the suggestions made by such officials. The key to these goals is working with our clients and suppliers by delivering the right products at the right time, which explains our products being displayed at stores across Gulf. To be in the World Class Quality retail solution provider to all Global Retail Giants in Home Furnishing segment with 100% Commitment. 24 Years of Excellence, N K International welcomes all buyers and importers to our twenty four year established Infrastructure. It has established itself in the Home Furnishing field since its inception in the year 1997. As we continue to grow, NK International is fully committed to excellence and prompt delivery of quality products across Gulf.

1.2 OBJECTIVE OF THE PROJECT :

One fundamental problem in sentiment analysis categorization of sentiment polarity. Given a piece of written text, the problem is to categorize the text into one specific sentiment polarity, positive or negative (or neutral). Based on the scope of the text, there are three distinctions of sentiment polarity categorization, namely the document level, the sentence level, and the entity and aspect level. Consider a review Ã¢â‚¬Å“I like multimedia features but the battery life sucks.Ã¢â‚¬This sentence has a mixed emotion. The emotion regarding multimedia is positive whereas that regarding battery life is negative. Hence, it is required to extract only those opinions relevant to a particular feature (like battery life or multimedia) and classify them, instead of taking the complete sentence and the overall sentiment. In this paper, we present a novel approach to identify pattern specific expressions of opinion in text.

CHAPTER – 2

2. SYSTEM STUDY

2.1 EXISTING SYSTEM

Existing approaches to sentiment analysis can be grouped into three main categories: knowledge-based techniques, statistical methods, and hybrid approaches. Knowledge-based techniques classify text by affect categories based on the presence of unambiguous affect words such as happy, sad, afraid, and bored. Some knowledge bases not only list obvious affect words, but also assign arbitrary words a probable "affinity" to particular emotions. Statistical methods leverage elements from machine learning such as latent semantic analysis, support vector machines, "bag of words", "Pointwise Mutual Information" for Semantic Orientation, semantic space models or word embedding models, and deep learning. More sophisticated methods try to detect the holder of a sentiment (i.e., the person who maintains that affective state) and the target (i.e., the entity about which the affect is felt). To mine the opinion in context and get the feature about which the speaker has opined, the grammatical relationships of words are used. Grammatical dependency relations are obtained by deep parsing of the text. Hybrid approaches leverage both machine learning and elements from knowledge representation such as ontologies and semantic networks in order to detect semantics that are expressed in a subtle manner, e.g., through the analysis of concepts that do not explicitly convey relevant information, but which are implicitly linked to other concepts that do so.

2.2.1 LIMITATION OF EXSISTING SYSTEM

Sentiment analysis systems trained on review data are often much less accurate when applied to data from other domains such as news or social media because of the differences in how people express themselves in these domains: journalists generally do not express sentiment in the same way a reviewer does, which are both different from how a poster on social media expresses sentiment. As a consequence, the patterns a machine learning system trained on review data has learnt to recognize as evidence for predicting sentiment in this domain will generally not be useful for predicting sentiment in other domains. Even more problematically, most online review data is in English. For global organizations, successful reputation management requires monitoring media sources in many languages.

2.2 PROPOSED SYSTEM

To overcome the drawbacks of the methods we have reviewed above, we propose a new model for sentiment analysis. In this model we combine many techniques to reach our final goal of emotion extraction. The steps for the process are documented below.

1. Retrieval of Data: data is mined using the existing for data extraction. Data's would be selected based on a few chosen keywords pertaining to the domain of our concern, i.e. product reviews. We have elected to use the API due to ease of data extraction.

2. Preprocessing: In this stage, the data is put through a preprocessing stage in which we remove identifying information such as data handles, timestamps of the message and embedded links and videos. Such information is largely irrelevant and may cause false results to be given by our system.

3. Word Correction: As text are written for human perusal, they often contain slang, misspellings and other irrelevant data. Thus we correct the misspellings in the sentences and look to replace the slang in the sentences with words from standard English that may roughly relate to the slang in question. As slang itself can be used to display a wide variety of sentiment, often with greater emotional impact, this process is necessary so that slang words may be considered as part of the emotion expressed.

4. Polarity detection: In this step we begin the second phase of our proposed system, in which we try to identify the polarity of the sentence in question. If emoticons exist in the statements, they will be used as well to compute the overall polarity of the statement. We aim to find sentences where the polarity detection is not very clear or where the expressed sentiment may be low. We also try to isolate the opinion words in the sentence in relation to a given concept in the sentence.

2.2.2 ADVANTAGES OF PROPOSED SYSTEM

That’s where sentiment analysis can help to:

- Understand what your customers like and dislike about your product.
- Compare your product reviews with those of your competitors.
- Get the latest product insights in real-time, 24/7.
- Save hundreds of hours of manual data processing.
- You can develop a more insightful, data-based marketing strategy
- Give extra boost to your customer service
- Find industry leaders and influencers and understand the customers.

2.3 HARDWARE SPECIFICATION

Processor:	AMD Ryzen 5000 series
RAM:	8 GB
Hard Disk Drive:	512 GB
Printer:	HP Ink Jet
Keyboard:	HP
Mouse:	Logi Tech (Optical)

2.4 SOFTWARE SPECIFICATION

Front End/GUI Tool:	HTML, CSS, Bootstrap
Operating System:	Windows 11
Coding language:	Python
Back End:	Python flask

2.5 SOFTWARE DESCRIPTION

Python :

python is an object-oriented programming language created by Guido Rossum in 1989. It is ideally designed for rapid prototyping of complex applications. It has interfaces to many OS system calls and libraries and is extensible to C or C++. Many large companies use the Python programming language, including NASA, Google, YouTube, BitTorrent, etc.

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python’s elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms. The Python interpreter and the extensive standard library are freely available in source or binary form for all major platforms from the Python Web site, https://www.python.org/, and may be freely distributed. The same site also contains distributions of and pointers to many free third-party Python modules, programs and tools, and additional documentation. The Python interpreter is easily extended with new functions and data types implemented in C or C++ (or other languages callable from C). Python is also suitable as an extension language for customizable applications. This tutorial introduces the reader informally to the basic concepts and features of the Python language and system. It helps to have a Python interpreter handy for hands-on experience, but all examples are self-contained, so the tutorial can be read off-line as well. For a description of standard objects and modules, see library-index. Reference-index gives a more formal definition of the language. To write extensions in C or C++, read extending-index and c-api-index. There are also several books covering Python in depth. This tutorial does not attempt to be comprehensive and cover every single feature, or even every commonly used feature. Instead, it introduces many of Python’s most noteworthy features, and will give you a good idea of the language’s flavor and style. After reading it, you will be able to read and write Python modules and programs, and you will be ready to learn more about the various Python library modules described in library-index.

Methods :

Methods on objects are functions attached to the object's class; the syntax instance. method(argument) is, for normal methods and functions, syntactic sugar for Class.method(instance, argument). Python methods have an explicit self-parameter to access instance data, in contrast to the implicit self (or this) in some other object-oriented programming languages (e.g., C++, Java, Objective-C, or Ruby).

Typing :

The standard type hierarchy in Python 3

Python uses duck typing and has typed objects but untyped variable names. Type constraints are not checked at compile time; rather, operations on an object may fail, signifying that the given object is not of a suitable type. Despite being dynamically-typed, Python is strongly-typed, forbidding operations that are not well-defined (for example, adding a number to a string) rather than silently attempting to make sense of them.

Python allows programmers to define their own types using classes, which are most often used for object-oriented programming. New instances of classes are constructed by calling the class (for example, SpamClass () or EggsClass ()), and the classes are instances of the metaclass type (itself an instance of itself), allowing metaprogramming and reflection.

Before version 3.0, Python had two kinds of classes: old-style and new-style.[96] The syntax of both styles is the same, the difference being whether the class object is inherited from, directly or indirectly (all new-style classes inherit from object and are instances of type). In versions of Python 2 from Python 2.2 onwards, both kinds of classes can be used. Old-style classes were eliminated in Python 3.0.

The long-term plan is to support gradual typing and from Python 3.5, the syntax of the language allows specifying static types but they are not checked in the default implementation, CPython. An experimental optional static type checker named mypy supports compile-time type checking.

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance.

Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed. Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace.

A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

Python Programming Characteristics :

- It provides rich data types and easier to read syntax than any other programming languages

- It is a platform-independent scripted language with full access to operating system API's

- Compared to other programming languages, it allows more run-time flexibility

- It includes the basic text manipulation facilities of Perl and Awk

- A module in Python may have one or more classes and free functions

- Libraries in Pythons are cross-platform compatible with Linux, Macintosh, and Windows

- For building large applications, Python can be compiled to byte-code

- Python supports functional and structured programming as well as OOP
- It supports interactive mode that allows interacting Testing and debugging of snippets of code
- In Python, since there is no compilation step, editing, debugging, and testing are fast.

Applications of Python Programming Language :

- Program video games
- Build Artificial Intelligence algorithms
- Program various scientific programs such as statistical models

Support vector classifier :

An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification, implicitly mapping their inputs into high-dimensional feature spaces. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier. Let you have basic understandings from this article before you proceed further. Here I'll discuss an example about SVM classification of cancer UCI datasets using machine learning tools i.e. scikit-learn compatible . **Pre-requisites:** Numpy, Pandas, matplot-lib, scikit-learn Let's have a quick example of support vector classification.

HTML :

HTML (Hyper Text Markup Language) is the most basic building block of the Web. It defines the meaning and structure of web content. Other technologies besides HTML are generally used to describe a web page's appearance/presentation (CSS) or functionality/behavior (JavaScript).

"Hypertext" refers to links that connect web pages to one another, either within a single website or between websites. Links are a fundamental aspect of the Web. By uploading content to the Internet and linking it to pages created by other people, you become an active participant in the World Wide Web.

HTML uses "markup" to annotate text, images, and other content for display in a Web browser. HTML markup includes special "elements" such as <head>, <title>, <body>, <header>, <footer>, <article>, <section>, <p>, <div>, <span>, <img>, <aside>, <audio>, <canvas>, <datalist>, <details>, <embed>, <nav>, <output>, <progress>, <video>, <ul>, <ol>, <li> and many others.

An HTML element is set off from other text in a document by "tags", which consist of the element name surrounded by "<" and ">". The name of an element inside a tag is case insensitive. That is, it can be written in uppercase, lowercase, or a mixture. For example, the <title> tag can be written as <Title>, <TITLE>, or in any other way.

CSS :

Cascading Style Sheets (**CSS**) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML or XML (including XML dialects such as SVG, MathML or XHTML). CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of content and presentation, including layout, colors, and fonts. This separation can improve content accessibility; provide more flexibility and control in the specification of presentation characteristics; enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, which reduces complexity and repetition in the structural content; and enable the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.

Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device.[4]

The name *cascading* comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.

The CSS specifications are maintained by the World Wide Web Consortium (W3C). Internet media type (MIME type) is registered for use with CSS by RFC 2318 (March 1998). The W3C operates a free CSS validation service for CSS documents.

In addition to HTML, other markup languages support the use of CSS including XHTML, plain XML, SVG, and XUL. CSS is also used in GTK widget toolkit.

Bootstrap :

Bootstrap is an HTML, CSS and JS library that focuses on simplifying the development of informative web pages (as opposed to web applications). The primary purpose of adding it to a web project is to apply Bootstrap's choices of color, size, font and layout to that project. As such, the primary factor is whether the developers in charge find those choices to their liking. Once added to a project, Bootstrap provides basic style definitions for all HTML elements. The result is a uniform appearance for prose, tables and form elements across web browsers. In addition, developers can take advantage of CSS classes defined in Bootstrap to further customize the appearance of their contents.

For example, Bootstrap has provisioned for light- and dark-colored tables, page headings, more prominent pull quotes, and text with a highlight.

Bootstrap also comes with several JavaScript components which do not require other libraries like jQuery. They provide additional user interface elements such as dialog boxes, tooltips, progress bars, navigation drop-downs, and carousels. Each Bootstrap component consists of an HTML structure, CSS declarations, and in some cases accompanying JavaScript code. They also extend the functionality of some existing interface elements, including for example an auto-complete function for input fields.

The most prominent components of Bootstrap are its layout components, as they affect an entire web page. The basic layout component is called "Container", as every other element in the page is placed in it. Developers can choose between a fixed-width container and a fluid-width container. While the latter always fills the width with the web page, the former uses one of the five predefined fixed widths, depending on the size of the screen showing the page :

- Smaller than 576 pixels
- 576–768 pixels

- 768–992 pixels
- 992–1200 pixels
- Larger than 1200 pixels

Once a container is in place, other Bootstrap layout components implement a CSS Flexbox layout through defining rows and columns.

A precompiled version of Bootstrap is available in the form of one CSS file and three JavaScript files that can be readily added to any project. The raw form of Bootstrap, however, enables developers to implement further customization and size optimizations. This raw form is modular, meaning that the developer can remove unneeded components, apply a theme and modify the uncompiled Sass files.

Python flask :

Flask is a web application framework written in Python. It was developed by Armin Ronacher, who led a team of international Python enthusiasts called Poocco. Flask is based on the Werkzeg WSGI toolkit and the Jinja2 template engine. Both are Pocco projects.

WSGI :

The Web Server Gateway Interface (Web Server Gateway Interface, WSGI) has been used as a standard for Python web application development. WSGI is the specification of a common interface between web servers and web applications.

Werkzeug :

Werkzeug is a WSGI toolkit that implements requests, response objects, and utility functions. This enables a web frame to be built on it. The Flask framework uses Werkzeg as one of its bases.

Jinja2 :

jinja2 is a popular template engine for Python.A web template system combines a template with a specific data source to render a dynamic web page.

Microframework :

Flask is often referred to as a microframework. It is designed to keep the core of the application simple and scalable. Instead of an abstraction layer for database support, Flask supports extensions to add such capabilities to the application.

2.6 MODULE DESCRIPTION :

1. Extraction of benchmark datas :

- The initial step in the process is to obtain review data from the source of benchmark data.
- The information is taken from postings, comments, reviews, and tweets.
- Twitter tweets, movie reviews, news feeds, product reviews, and Facebook postings are some of the sources of frequently utilized datasets.
- The extracted data are fed into the system at this step, which is used for data mining and analysis.
- This stage serves as the central component of the sentiment analysis process.

2. Data preprocessing :

- Data preprocessing is a crucial phase in text data analysis.
- Due to the repetitions and redundancies in tweets, blogs, reviews, and other types of text, text data become more complicated.
- Data normalization uses data preprocessing as a filtering method.
- The normalization, word tokenization, removing stop words, removing extra spaces, padding, changing the text data to lowercase, and removing hash tagging are examples of data preprocessing, etc.
- This work implemented various tasks to achieve the data in the desired format.

3. Feature encoding :

- Feature Encoding for Numerical Representation of Textual Data The obtained datasets might not be in a format suitable for statistical or mathematical calculations.
- A proper function encoding method is needed to extract numerical characteristics from available text data.
- We must propose a mathematical model which correctly depicts each review in the sample and captures the accurate or true semanticist word or sentence therein.

4. Sentiment prediction :

- Predicting sentiments from the supplied input data is helpful in this process stage.
- Several process cycles may be necessary for the algorithms to become more generic.
- The sentiment prediction’s findings and the sentiment’s outcomes are connected.
- It boosts the sentiment analysis’s productivity.

5. Sentiment analysis :

- We can define the polarity of the texts after all the above-described stages as analysts.
- The analytical results are listed in this step.
- The words can be positive or negative.
- Each performance measure is defined in the following way.

6. SVC model :

Support Vector Machine(SVM) is a supervised machine learning algorithm used for both classification and regression. Though we say regression problems as well it’s best suited for classification. The objective of the SVM algorithm is to find a hyperplane in an N-dimensional space that distinctly classifies the data points. The dimension of the hyperplane depends upon the number of features. If the number of input features is two, then the hyperplane is just a line. If the number of input features is three, then the hyperplane becomes a 2-D plane. It becomes difficult to imagine when the number of features exceeds three.

7. SVC Model Algorithm Sample Steps :

```
import pandas as pd
import numpy as np
df=pd.read_excel("D:/ScrubbedDataCorrected1.xlsx")
df["Comments"].fillna(value = "", inplace = True)
df.dropna(subset=['Rating', 'Comments'], inplace=True)
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.svm import SVC
model=SVC()
from sklearn.pipeline import make_pipeline
text_model=make_pipeline(CountVectorizer(),SVC())
```

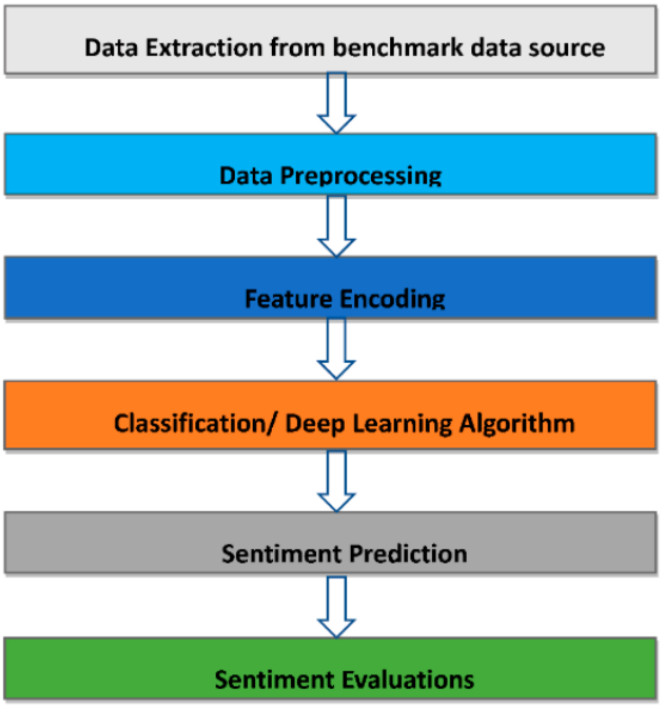
```
text_model.fit(x_train,y_train)
y_pred=text_model.predict(x_test)
from sklearn.metrics import accuracy_score
print(accuracy_score(y_pred,y_test))
import joblib
joblib.dump(text_model,'Project')
import joblib
```

CHAPTER 3

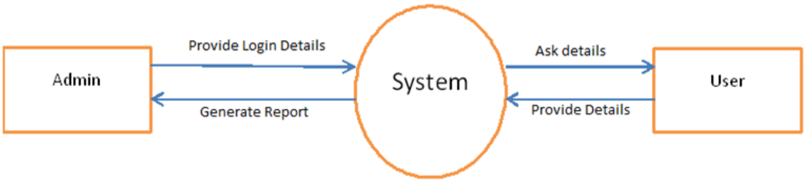
3. SYSTEM DESIGN & DEVELOPMENTS

3.1 DATA FLOW DIAGRAM

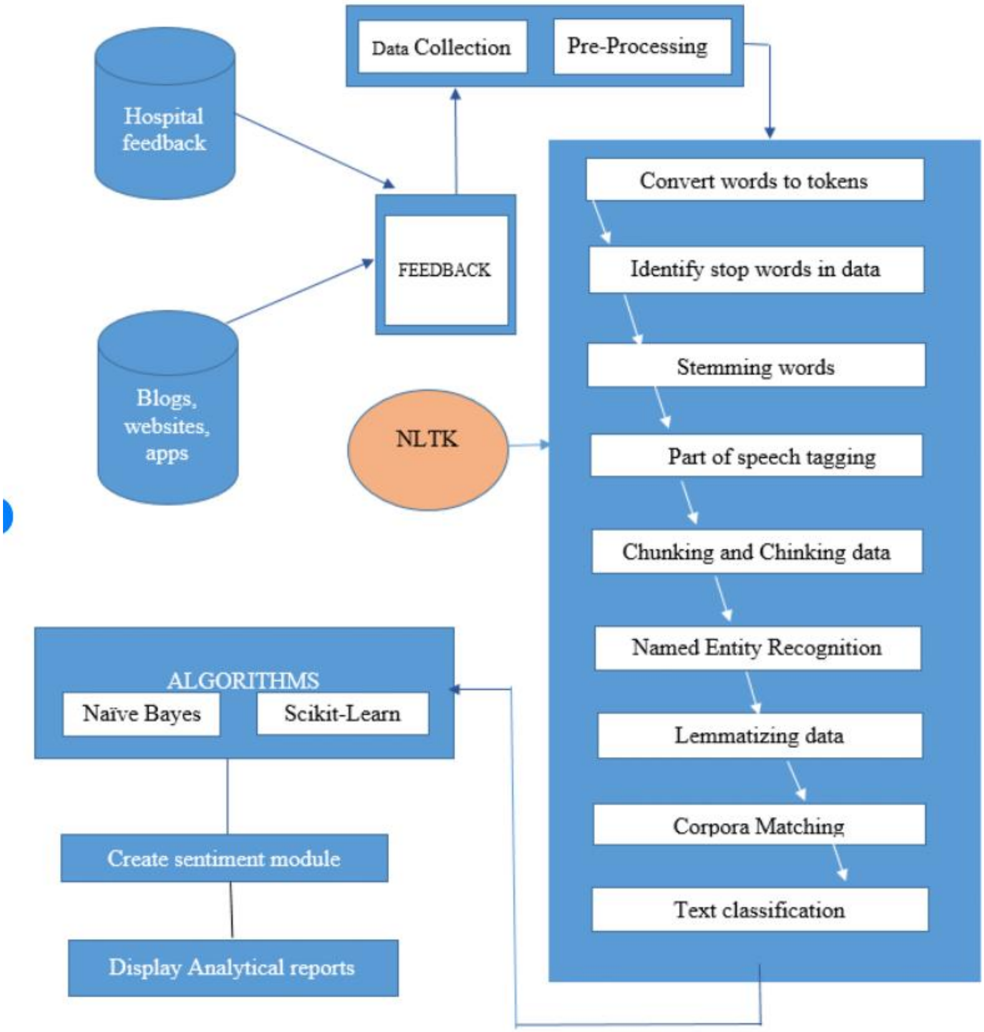
LEVEL 1


Data Extraction from benchmark data source
Data Preprocessing
Feature Encoding
Classification/ Deep Learning Algorithm
Sentiment Prediction
Sentiment Evaluations

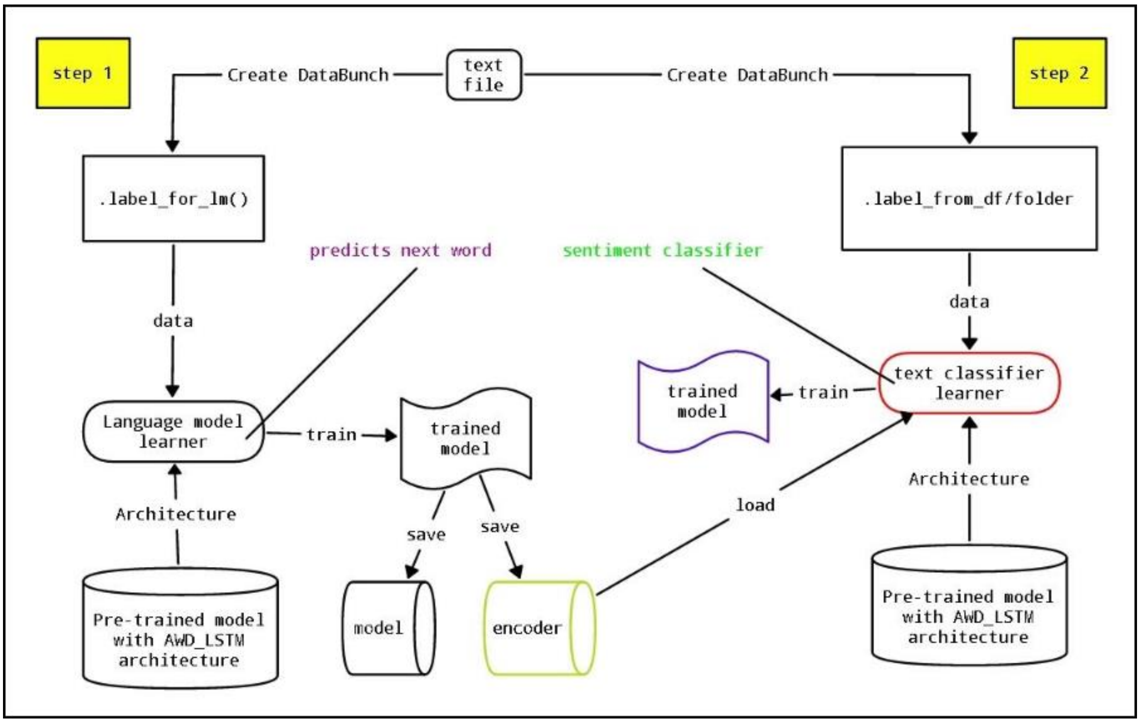
LEVEL 2



3.2 ENTITY FLOW DIAGRAM



3.3 SYSTEM FLOW DIAGRAM



3.4 INPUT DESIGN

Input Design converts the user-oriented inputs to computer-based formats. Inaccurate input data are the most common cause of errors in data processing. Error data entered by the data operator can be controlled by the input design. The goal of designing input is to make the data entry easy, logical and as free from errors as much as possible.

The proposed system is completely menu-driven. It is a powerful tool for interactive design. It helps the user comprehend the range of alternatives available and also prevents them from making an invalid selection. All entry screens are interactive in nature. It has been designed taking into account all the constraints of the end-user.

Some other features included are:

- The form title clearly states the purpose of the form
- Adequate space is given for data entry

  Data Validation is done for eliminating duplicate entries

3.5 OUTPUT DESIGN

Outputs are the most important and direct source of information to the customer and management. Intelligent output design will improve the system's relationship with the user and help in decision making. Outputs are used to make permanent hard copy of the results for later consultation. The output generated by the system is often regarded as the criteria for evaluating the performance of the system. The output design was based on the following factors.

- Usefulness determining the various outputs to be printed to the system user.
- Differentiating between the outputs to be displayed and those to be printed.
- The format for the presentation of the output.

For the proposed system, it is necessary that the output should be compatible with the existing manual reports. The outputs have been formatted with this consideration in mind. The outputs are obtained after all the phase, from the system can be displayed or can be produced in the hard copy. The hard copy is highly preferred since it can be used by the controller section for future reference and it can be used for maintaining the record.

3.6 CODE DESIGN

Code is an ordered collection of symbols designed to provide unique identification of an attribute. Codes can be used for various purposes. They can specify object’s physical or performance characteristics and they can be used to give operational instructions. They also can show inter relationships and may sometimes use to achieve secrecy or confidentiality. Codes are designed for optimum human-oriented use and machine efficiency. Codes posses’ uniqueness, expandability, conciseness, uniform nets, simplicity, versatility, sort ability, meaningfulness and operability.

Sufficient effort and time is spent in the preliminary study of the problem to design an efficient code. Activate serve scripting is object oriented. The source code is designed so that it can do transaction efficiently. It is the code that dose all the updating, modifications, etc. for all object used in the project there exist an associated source code, which explains the work of that object. It also describes the flow of the project.

Source code is enhanced by structured coding techniques by good internal comments and features provided by the language.

The code design in this project is made modular. The modular behavior enables easy debugging and testing. Inserting comment statement wherever enhances the coding. This is done during the documentation process coding is done in such a way that errors can be trapped easily. Also modifications can easily be appended due to the codes modular behavior.

CHAPTER 4

4. SYSTEM TESTING

4.1 UNIT TESTING

Unit testing focuses verification efforts on the smallest unit of software design, the module. This is also known as “Module Testing” The modules are tested separately this testing is carried out during programming stage itself. In this step each module is found to be working satisfaction as regard to the expected output from the module.

4.2 INTEGRATION TESTING

Integration testing focuses on the design and construction of the software architecture. Data can be lost across an interface; one module can have adverse effect on another sub functions and show on. Thus, integration testing is a systematic technique for constructing test to uncover errors associated with in the interface. In this project, all the modules are companied and then the entire program is tested as a whole.

4.3 VALIDATION TESTING

Validation testing is the requirement established as a part of software requirement analysis is validated against the software that has been constructed. This test provides the final assurance whether the software needs all functional, behavioral and performance requirements

Thus the proposed system under consideration has been tested by using validation testing and found to be working satisfactory.

CHAPTER 5

5. SYSTEM IMPLEMENTATION AND MAINTENANCE

Implementation is the stage where the theoretical design is turned into a working system. The most crucial stage in achieving a new successful system and in giving confidence on the new system for the users that it will work efficiently and effectively. The system can be implemented only after thorough testing is done and if it is found to work according to the specification. It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the change over and an evaluation of change over methods a part from planning. Two major tasks of preparing the implementation are education and training of the users and testing of the system.The more complex the system being implemented, the more involved will be the systems analysis and design effort required just for implementation. The implementation phase comprises of several activities. The required hardware and software acquisition is carried out. The system may require some software to be developed. This application is implemented in python as front end mysql as back end.

System Maintenance:

The maintenance plan specifies the scheduled servicing tasks and intervals (preventive maintenance) and the unscheduled servicing tasks (adaptive or corrective maintenance). Tasks in the maintenance plan are allocated to the various maintenance agencies. A maintenance allocation chart is developed to tag the maintenance tasks to the appropriate maintenance agencies. These include: in-service or in-house work centers, approved contractors, affiliated maintenance or repair facilities, original equipment manufacturer, etc. The maintenance plan also establishes the requirements for the support resources.

Related activities such as resource planning, budgeting, performance monitoring, upgrades, longer term supportability, and sustenance also need to be managed. These activities are being planned, managed, and executed over a longer time horizon and they concern the well being of the system over the entire life cycle. Proper maintenance of the system (including maintenance-free system designs) relies very much on the availability of support resources, such as support and test equipment, technical data and documentation, personnel, spares, and facilities.

CHAPTER 6

6. CONCLUSION

The era of digitization marks the astonishing growth of subjective textual data online. Proper analysis of the textual information, to rightly reflect the public sentiment regarding any topic, demands proper investigation of textual data. Sentiment analysis has emerged as the most important task which helps to enhance the decision-making process by extracting the underlying sentiment or opinion of data. Even though sentiment analysis has progressed in recent years, modern models have flaws such as domain dependence, negation management, high dimensionality, and the failure to use efficient keyword extraction. This paper examines and provides a comprehensive discussion of different perspectives related to the creation and implementation of an effective sentiment analysis model. A thorough examination and establishment of various modules of the sentiment analysis methodology are carried out to plan and improve effective sentiment analysis models. The keyword extraction algorithm is vital to the success of a sentiment analysis model and thus is well-studied in this paper. The paper also discusses sentiment classification methods, which form an essential aspect of a sentiment analysis model. The paper conducts a detailed review of both machine learning and lexicon-based approaches to textual data sentiment analysis. As a thorough, well-organized study on sentiment analysis, this research effort can assist academicians and industry experts in analyzing and developing powerful sentiment analysis models in a wide range of domains. Sentiment analysis models have a lot of potential for further development and use in the near future because they have a broad range of uses in social, industrial, political, economic, health and safety, education, defense financial contexts, and others. Each of the sentiment analysis modules as discussed in this paper can be investigated, improvised, and supplemented with certain relevant algorithms to design an efficient sentiment analysis model. This study also offers prospective guidelines for carrying out proper sentiment analysis research

CHAPTER 7

7. FUTURE SCOPE

Sentiment analysis is a uniquely powerful tool for businesses that are looking to measure attitudes, feelings and emotions regarding their brand. To date, the majority of sentiment analysis projects have been conducted almost exclusively by companies and brands through the use of social media data, survey responses and other hubs of user-generated content. By investigating and analyzing customer sentiments, these brands are able to get an inside look at consumer behaviors and, ultimately, better serve their audiences with the products, services and experiences they offer.

The future of sentiment analysis is going to continue to dig deeper, far past the surface of the number of likes, comments and shares, and aim to reach, and truly understand, the significance of social media interactions and what they tell us about the consumers behind the screens. This forecast also predicts broader applications for sentiment analysis – brands will continue to leverage this tool, but so will individuals in the public eye, governments, nonprofits, education centers and many other organizations.

CHAPTER 8

8. BIBILIOGRAPHY

1. Abdul-Mageed, M., M.T. Diab, and M. Korayem. Subjectivity and sentiment analysis of modern standard Arabic. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics:shortpapers, 2011.

2. Akkaya, C., J. Wiebe, and R. Mihalcea. Subjectivity word sense disambiguation. In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP-2009), 2009.

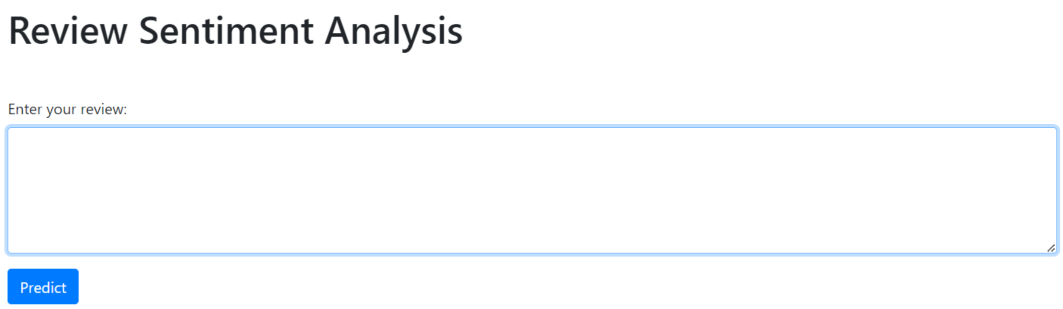
3. Alm, C.O. Subjective natural language problems: motivations, applications, characterizations, and implications. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics:shortpapers (ACL-2011), 2011.

4. Andreevskaia, A. and S. Bergler. Mining WordNet for fuzzy sentiment: Sentiment tag extraction from WordNet glosses. In Proceedings of Conference of the European Chapter of the Association for Computational Linguistics (EACL-06), 2006.

5. Archak, N., A. Ghose, and P.G. Ipeirotis. Show me the money!: deriving the pricing power of product features by mining consumer reviews. In Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD-2007), 2007.

APPENDIX

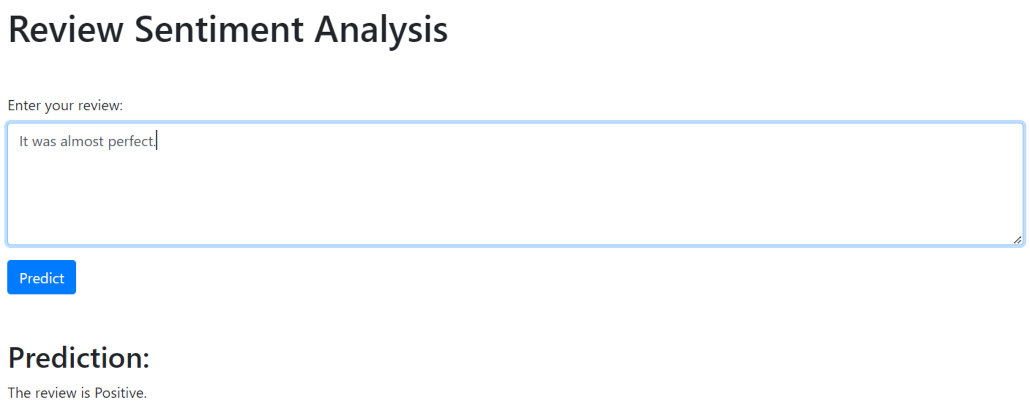
a. Form design :



Review Sentiment Analysis

Enter your review:

Positive form design :



Review Sentiment Analysis

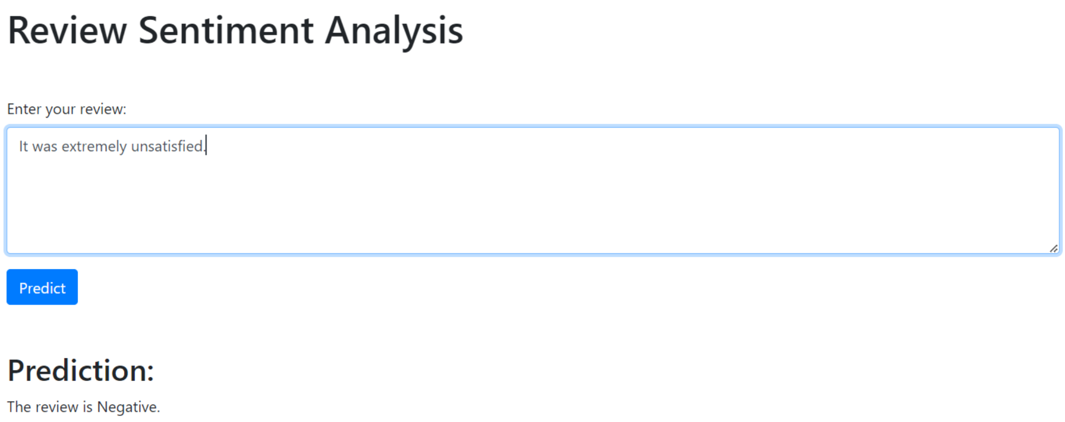
Enter your review:

It was almost perfect.

Prediction:

The review is Positive.

Negative form design :



Review Sentiment Analysis

Enter your review:

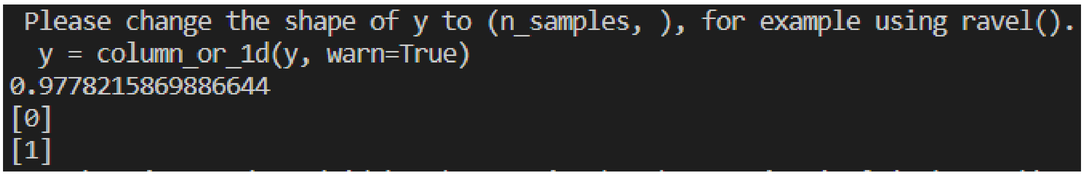
It was extremely unsatisfied.

Prediction:

The review is Negative.

Prediction chart :

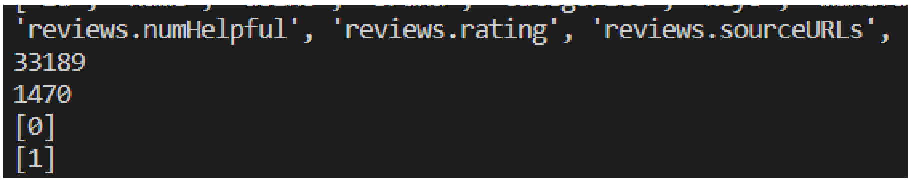
Chart for training the sample data set :



```
 Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
0.9778215869886644
[0]
[1]
```

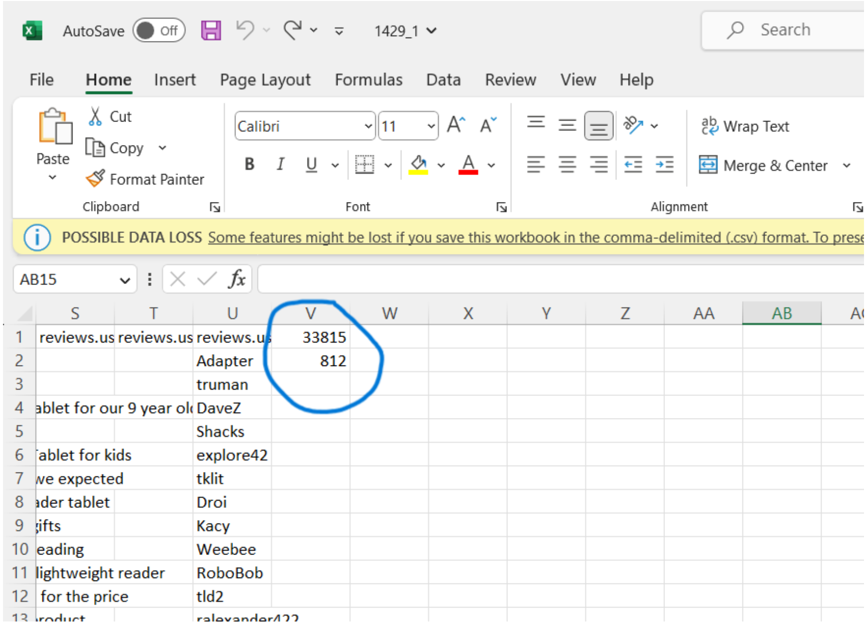
The above chart shows the results for assuming the sample data set.

Chart for actual data set of reviews :



```
'reviews.numHelpful', 'reviews.rating', 'reviews.sourceURLs',
33189
1470
[0]
[1]
```

The above chart predicts the actual number of positive and negative reviews in the data set which is almost similar to the Microsoft excel's math function calculations.



POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To prese

	S	T	U	V	W	X	Y	Z	AA	AB
1	reviews.us	reviews.us	reviews.u	33815						
2			Adapter	812						
3			truman							
4	ablet for our 9 year ol		DaveZ							
5			Shacks							
6	ablet for kids		explore42							
7	we expected		tklit							
8	ader tablet		Droi							
9	ifts		Kacy							
10	eading		Weebee							
11	lightweight reader		RoboBob							
12	for the price		tld2							
13	roduct		ralexander422							

b. Sample code :

```
import pandas as pd
import numpy as np
df=pd.read_excel("D:/ScrubbedDataCorrected1.xlsx")
df["Comments"].fillna(value = "", inplace = True)
df.dropna(subset=['Rating', 'Comments'], inplace=True)
summary=[]
for i in df['Rating']:
  if i<=2:
    summary.append(0)
  else:
    summary.append(1)
x=df['Comments'].values.astype('U')
y=pd.DataFrame(summary)
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,random_state=0)
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.svm import SVC
model=SVC()
from sklearn.pipeline import make_pipeline
text_model=make_pipeline(CountVectorizer(),SVC())
text_model.fit(x_train,y_train)
y_pred=text_model.predict(x_test)
from sklearn.metrics import accuracy_score
print(accuracy_score(y_pred,y_test))
import joblib
joblib.dump(text_model,'Project')
import joblib
text_model=joblib.load('Project')
print(text_model.predict(["It was really bad"]))
print(text_model.predict(["It was the best lens i have ever used!"]))
```

Implementation :

```
import joblib
text_model = joblib.load('Project')
import pandas as pd
df = pd.read_csv("D:/new dataset/1429_1.csv", delimiter=",")
positive = 0
negative = 0
neutral = 0
print(df.columns.tolist())
df.dropna(subset=['reviews.text'], inplace=True)
for i in df['reviews.text']:
    if text_model.predict([i])[0]==1:
        positive += 1
    else:
        negative += 1
print(positive)
print(negative)
print(text_model.predict(["It was really bad"]))
print(text_model.predict(["It was the best lens i have ever used!"]))
```

Flask :

```
from flask import Flask, render_template, request
import joblib
import numpy as np

app = Flask(__name__)
model = joblib.load('Project')

@app.route('/')
def home():
    return render_template('index.html')

@app.route('/predict', methods=['POST'])
def predict():
    review = request.form['review']
    prediction = model.predict([review])[0]
    if prediction == 0:
        sentiment = 'Negative'
    else:
        sentiment = 'Positive'
    return render_template('index.html', prediction_text='The review is
{}.'.format(sentiment))

if __name__ == '__main__':
    app.run(debug=True)
```

NAME : BARANI.S

ROLL NO : 20BCS007

EMAIL : baranis20bcs007@skasc.ac.in

ADDRESS : 6087/154, Rani rama devi nagar,
Sheshaiya sastri road,
Pudukkottai-622003

PROJECT DESCRIPTON

Sentiment analysis or opinion mining is one of the major tasks of NLP (Natural Language Processing). Sentiment analysis has gain much attention in recent years. In this project, we aim to tackle the problem of sentiment polarity categorization, which is one of the fundamental problems of sentiment analysis. A general process for sentiment polarity categorization is proposed with detailed process descriptions. Data used in this study are online product reviews collected from Amazon.com. Experiments for both sentence-level categorization and review-level categorization are performed with promising outcomes. At last, we also give insight into our future work on sentiment analysis. Sentiment is an attitude, thought, or judgment prompted by feeling. Sentiment analysis, which is also known as opinion mining, studies people's sentiments towards certain entities. Internet is a resourceful place with respect to sentiment information.